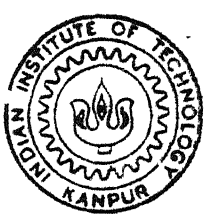
9210549

ASPECTS OF PATH PLANNING WITH SNELL'S LAW

by

SANJAY P. SHARMA

ME
TH
me/1994/M
Sh23a
1994
M
SHA
ASD



DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

May, 1994

ASPECTS OF PATH PLANNING WITH SNELL'S LAW

A thesis submitted
in partial fulfilment of the requirements
for the degree of

MASTER OF TECHNOLOGY

by

SANJAY P. SHARMA

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY-KANPUR
May-1994

30 MAY 1994 | ME
CENTRAL LIBRARY
I I T KANPUR
Acc No. A. 117813

ME-1994-M-SHA-ASP

CERTIFICATE

This is to certify that the thesis entitled **Aspects of Path Planning with Snell Law**, by *Sanjay P. Sharma* has been carried out under my supervision and that th work has not been submitted elsewhere for a degree.

(Dr. Amit Mukerjee)
Associate Professor
Dept. of Mechanical Engineering
Indian Institute of Technology
Kanpur 208016

ACKNOWLEDGEMENTS

I am greatly indebted to my thesis supervisor, Dr. Amit Mukerjee, for his guidance throughout the preparation of this thesis. Accomplishment of the present work would not have been possible without his sustained encouragement and open hearted cooperation.

I am thankful to my friends, Sudip, Ram Bhushan and Khurshid, for their valuable help at various stages of this work.

(Sanjay P. Sharma)

ABSTRACT

Maps, as treated in traditional Robot Motion Planning are not always black and white. Any terrain map can be divided into different regions depending upon the negotiability in the terrain. For some regions the negotiability may be so poor that if start and destination are outside it, it should always be avoided for Path Planning purposes. In this work an algorithm is presented to find such convex polygonal regions termed as *True Obstacles*. Such knowledge makes the path planning approach simpler.

There is an exact analogy between light rays and minimum cost paths at the boundaries of uniform cost regions. The algorithm for determining true obstacles is based on Snell's Law. We also present a search algorithm to find the minimum cost path which passes through a given sequence of boundary segments.

Contents

1 Introduction 1

2 True Obstacles 6

2.1 Nomenclature 6

2.2 Maximum Chord-ratio Theorem 9

2.3 Verifying a True Obstacle 16

2.4 Genetic Algorithms 19

2.5 Results 20

3 Snell's Law In Path Planning 23

3.1 Minimum Cost Path 23

3.2 Snell's Law and Path Planning 25

3.3 Multiregion Snell's Law 28

3.4 Multiregion Snell's Law Algorithm 31

4 Conclusions 36

List of Figures

1.1 A map of general terrain showing its nature 2

1.2 A slope map of general terrtain 2

1.3 Terrain Obstacle - Here, for any start and goal outside the region marked "Hills", an optimal path can never go through it, due to its high cost. Such a region is called a *True Obstacle*. 5

1.4 Movable Obstacle - To reach G from S, the stool can be moved (dashed line) or one can go around. The cost of moving some objects (e.g., the piano) is so highthat for all practical purposes it may be considered to be a true obstacle. 5

2.1 Nomenclature - Explains the terms used in the maximum chord-ratio theorem and the true obstacle algorithm. 7

2.2 Proposition 1 - λ_{max}(R) cannot occur at a vertex of Region R - For $x \rightarrow 0, \lambda$(S, G) is smaller than at least one of λ(S′, G′) and λ(S″, G″). . 11

2.3 Proposition 2 - λ_{max} cannot occur at a pair of points S and G, if θ(S, G) $\neq \theta$(G, S) - if θ(G, S) $> \theta$(S, G), λ(S′, G′) $> \lambda$(S, G), for $x \rightarrow 0$. Similar is the case when θ(S, G) $> \theta$(G, S). 11

2.4 Proposition 3 - λ_{max}(R) cannot occur at a pair of points S and G, if θ(S, G) $< \theta_{Cr}$(S, G) - In such a case, we can always choose a point G′ in the direction of reducing peri-dist as shown, such that λ(S, G′) $> \lambda$(S, G). 13

2.5 Proposition 4 - Given θ(S, G) $> \theta_{Cr}$(S, G), λ_{max}(R) cannot occur at the pair of points S and G if ρ(S, G) $< \rho_{max}$(R) - In such a case, we can always choosea point G′ in the direction of increasing peri-dist as shown, such that λ(S, G′) $> \lambda$(S, G). 14

2.6 Proposition 5 - If $\theta(S, G) = \theta(G, S) = \theta_{Cr}(S, G)$, there is a set of parallel chords with same chord-ratio. If this set does not include the chord between two perimetric image points, $\lambda_{max}(R)$ cannot occur at this pair of points. 15

2.7 Example polygon - Finding λ_{max} of a convex polygon. 18

2.8 Graph of time complexity for the true obstacle algorithm. The linear nature complies with complexity analysis result. 21

2.9 Some sample randomly generated convex polygons tested by the true obstacle algorithm. Chord shown is between perimetric image points for which the chord-ratio is maximum ($\lambda_{max}(R)$). 21

2.10 Some more sample convex polygons tested by the true obstacle algorithm. Chord shown is between perimetric image points for which the chord-ratio is maximum ($\lambda_{max}(R)$). 22

3.1 Minimum cost path {S, P_1, P_2, ..., P_{N-1}}which passes through the boundaries b_1, b_2, ..., b_{N-1} of regions R_1, R_2, ..., R_N with costs C_1, C_2, ..., C_N respectively. 24

3.2 Snell's Law of refraction - Light ray refracts at the boundary of media A and B with indices of refraction μ_A and μ_B obeying the ralation $\mu_A \sin\theta_A = \mu_B \sin\theta_B$. 26

3.3 Snell's Law obeying path at the boundary of two regions is the minimum cost path. 26

3.4 Mirror boundaries - Although the minimum cost path is {S, P, G}, light will not be able to enter the low cost region, as the boundaries act like mirros. 28

3.5 The path followed by light (dashed line) has a higher cost than the path {S, P_1, P_2, P_3, P_4, G}which follows the contour in the low cost region. . 29

3.6 The path {S, Q_1, Q_2, G}has a higher cost than the path {S, P_1, P_2, G}. 29

3.7 To find the minimum cost path between S and G in convex polygonal regions, which is known to pass through a sequence of edges. 34

3.8 To find the minimum cost path between S and G in concave polygonal regions, which is known to pass through a sequence of edges. 35

Chapter 1

Introduction

In traditional robot motion planning [Latombe], regions have been treated in black and white i.e., either obstacle or free space. The problem of path planning in such black and white regions has been studied extensively. However, in practice we come across regions having finite costs [Mitchell], [Rowe and Richbourg]. The problem of finding minimum time path through general terrain faced by a tank is one such example. Here, several types of informations are available, for example, the nature of terrain : forest, river, scrub, sand etc. (Fig. 1.1) or the lay of the land i.e., slopes along the path (Fig. 1.2). Even in ordinary mobile robot traversal, the mobile robot maintains a probabilistic, geometric map of the surroundings using sensors. Consequently the emerging map is not black and white, (e.g., Certainty Grid [Moravec]), though the regions here, are in black and white, the incompleteness of the robot's knowledge makes the cost of the region variable.

Finite costs actually can be asssigned to even traditional "obstacles" even a piano can be moved out of the path when installing new furniture, say. However, the assumption used in traditional motion planning is that for most obstacles, under normal navigation conditions, such costs are very high as to make it infeasible to attempt it. One of the main objectives of this work is to investigate the limitations of this assumption. The question we ask is: under what conditions is it valid to assume that an obstacle is better bypassed? Such obstacles, which we call "True Obstacles", are one of the main issues discussed in this work.

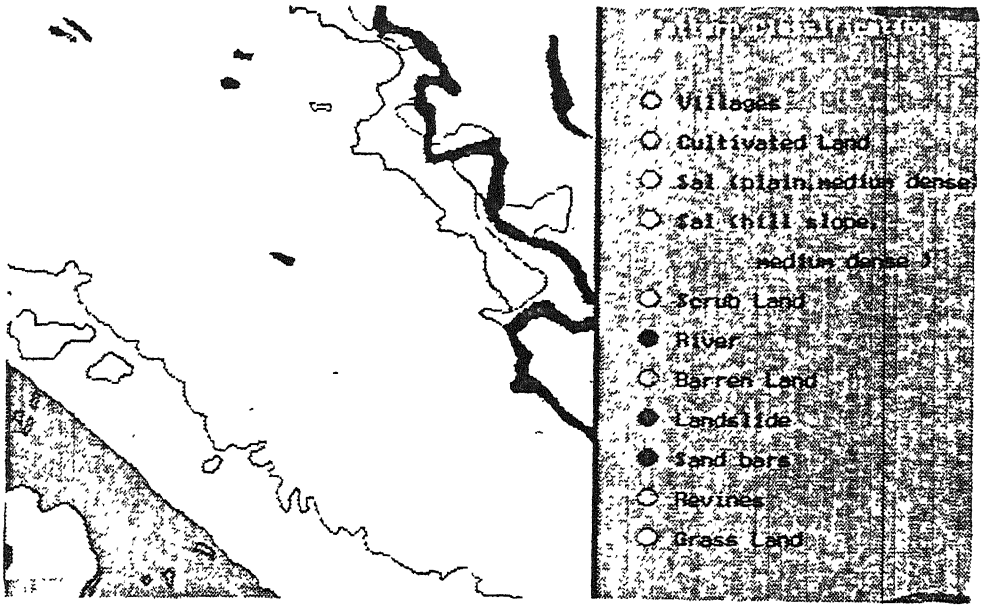


Figure 1.1: A map of general terrain showing its nature

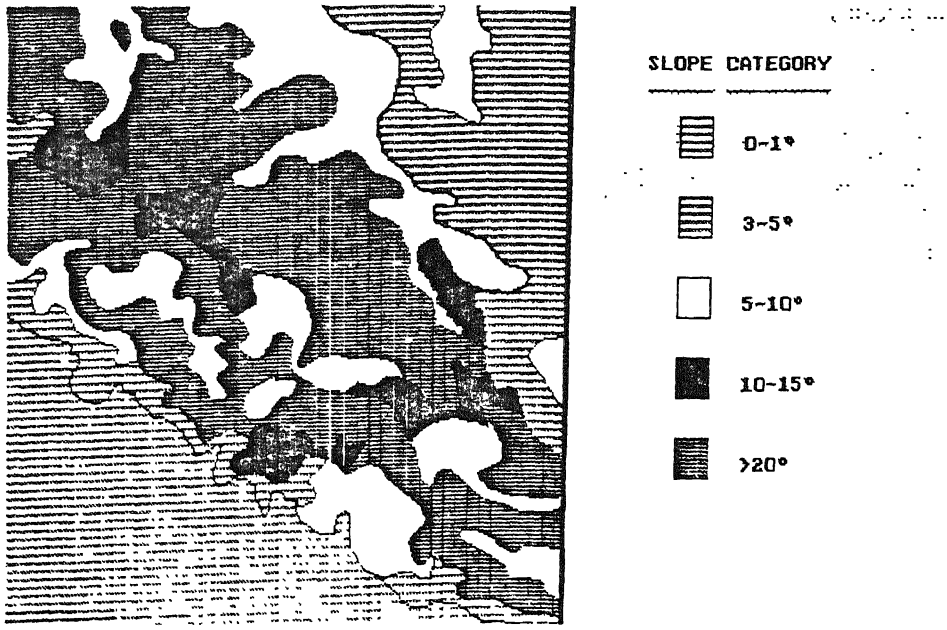


Figure 1.2: A slope map of general terrtain

The navigation problem where different costs are associated with different regions is termed as weighted region problem. This is a generalization of the well-studied obstacle avoidance problem, as the obstacle avoidance problem is simply the weighted region problem in which the costs or weights are either 1 or ∞ depending on whether a region is free space or obstacle. Multiple terrain costs are essential to model any outdoor terrain. How costs should be assigned to regions is a complex problem, and is largely application-dependent.

The model of cost used in some applications is the reciprocal of maximum speed in each region, where minimizing the cost of the path means minimizing the time required for a vehicle to execute the path. The cost of a region may change with the vehicle, the task, and other attributes of the problem such as fuel left, weather, etc. The variable cost problem has been studied mostly in the context of terrain navigation.

Negotiability of the vehicle in a region decides the cost of the region. Soil strength and slope influence the negotiability in a region. Soil strength depends on the soil type, soil depth and moisture content. Slope has a direction which makes the cost anisotropic and hence is quite difficult to handle. In military and nuclear applications, there may be regions which correspond to the high risk, either from enemy threat or due to radiation exposure. Costs can be assigned to traveling in these risky regions as well. Higher cost represents a poor negotiability in the region.

To simplify the complexities of modelling the full terrain data, most models [Mitchell], [Rowe and Richbourg] approximate the terrain with uniform isotropic cost regions. In this work also, we have assumed that the map can be decomposed into uniform isotropic cost regions. Further, we assume that the terrain features can be modelled using a polygonal representation [Rowe and Richbourg]. This is a widely used approximation in many domains. Hence, it is assumed that the map can be decomposed into polygonal regions with uniform isotropic costs.

When a map is represented as polygonal regions with uniform isotropic costs, sometimes the cost of a region is so high that no outside path should enter it. Such regions are called *True Obstacles*. The two undersaid examples make the concept of true obstacles more limpid.

Example 1 : Consider a polygonal approximation (Fig. 1.3) of a part of the general terrain given in Fig. 1.1. Here, for any start and goal outside the region marked "Hills", an optimal path can never go through it, due to its high cost. Such a region is called a *True Obstacle.*

Example 2 : For reaching goal point G (Fig. 1.4) from start point S, the stool can be moved (dashed line) or one can go around. Under normal circumstances, it might be pragmatic to move the stool. But if a heavy flowerpot is kept on the stool, it might be desirable to avoid moving the stool, unless the objective is to move the stool itself. Stool with heavy flowerpot, thus, becomes a true obstacle. The cost of moving some objects (e.g., the piano) is so high that for all practical purposes it may be considered to be a true obstacle. Thus, if costs can be assigned, such problems can also be handled by the concept of true obstacles.

Chapter 2 deals with true obstacles in more details. An algorithm to find the condition under which a region becomes a true obstacle is also presented in this chapter.

There is an exact analogy between the behaviour of light rays at the boundary of two media and minimum cost path at the boundary of two homogeneous cost regions. Chapter 3 discusses how this analogy can be utilized to find the minimum cost path in a map represented as uniform cost regions.

Conclusion of the present work is elucidated in chapter 4. Scope for extending the present work is also discussed in this chapter.

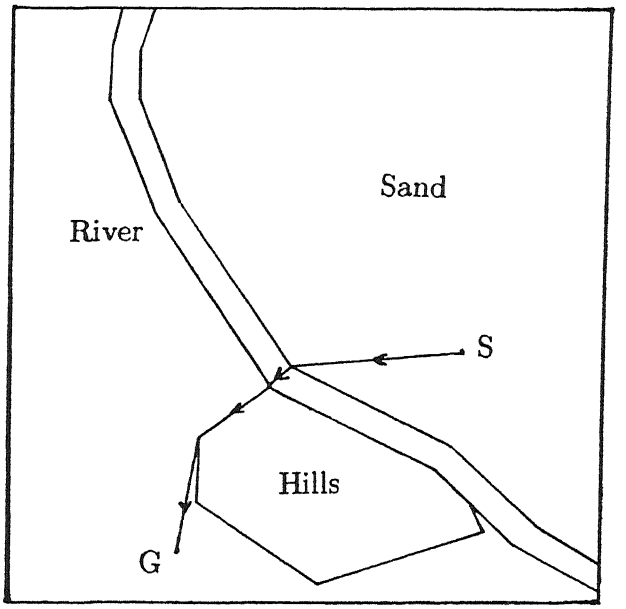


Figure 1.3: Terrain Obstacle - Here, for any start and goal outside the region marked "Hills", an optimal path can never go through it, due to its high cost. Such a region is called a *True Obstacle.*

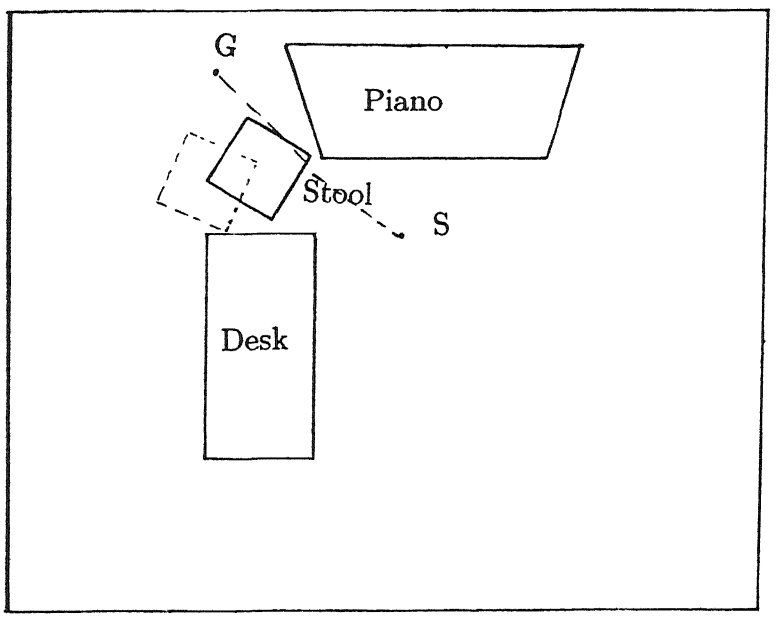


Figure 1.4: Movable Obstacle - To reach G from S, the stool can be moved (dashed line) or one can go around. The cost of moving some objects (e.g., the piano) is so high that for all practical purposes it may be considered to be a true obstacle.

Chapter 2

True Obstacles

For the purpose of path planning, a given map can be divided into different regions depending on the cost of moving. Sometimes the cost of moving in a region is so high that no outside path should enter it. Such regions are called True Obstacles. When start and goal points are outside such regions, these act like obstacles, even though finite costs are associated with such regions. Finding true obstacles in a map simplifies the path planning problem. In this chapter a computational geometry approach for finding true obstacles is presented. The algorithm proposed applies to convex polygonal regions and finds true obstacles.

2.1 Nomenclature

Following terms are used in this chapter (Fig. 2.1).

- Convex perimeter :- For a region, the perimeter of the smallest convex polygon enclosing the region is its convex perimeter.
- Convex rope length^{ccw}(ρ_{ccw}) :- For any two points S and G on the perimeter of a region, the rope length joining S and G in counter clock wise direction along the convex perimeter of the region is ρ_{ccw}(S, G) {S-P_1-P_2-G}.

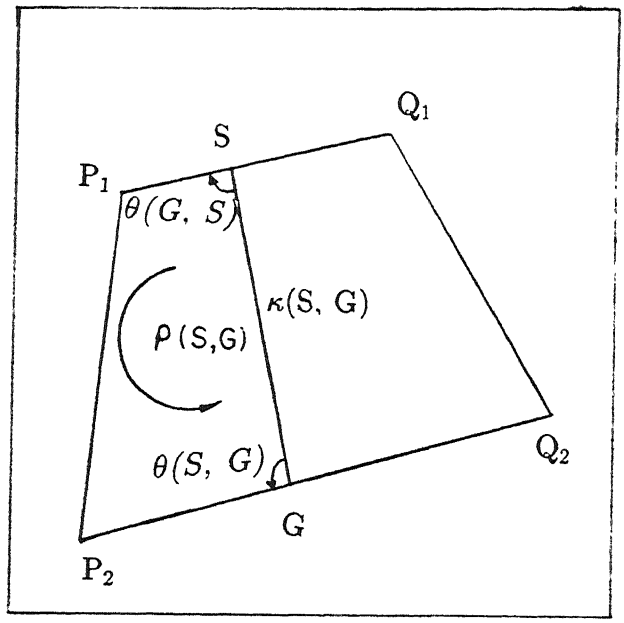


Figure 2.1: Nomenclature - Explains the terms used in the maximum chord-ratio theorem and the true obstacle algorithm.

- Convex rope length^{cw}(ρ_{cw}) :- For any two points S and G on the perimeter of a region, the rope length joining S and G in clock wise direction along the convex perimeter of the region is ρ_{cw}(S, G) {S-Q_1-Q_2-G}.

- Peri-dist (ρ) :- For any two points S and G on the perimeter of a region, ρ(S, G) is the minimum distance from S to G along the convex perimeter of the region. Thus

$$\rho(S, G) = \text{Min}(\rho_{ccw}(S, G), \rho_{cw}(S, G))$$

- Maximum peri-dist (ρ_{max}) :- If for any two points S and G on the perimeter of a region ρ_{ccw} equals ρ_{cw}, the ρ(S, G) becomes maximum. Thus, ρ_{max} for a region is half its convex perimeter.

- Perimetric image :- For any point P on the perimeter of a region, the point at a distance ρ_{max} from the point P along the convex perimeter of the region is called the perimetric image of P.

- Antipodal edge :- For any edge E of a convex polygon, the edge containing perimetric image of first vertex of E is called the antipodal edge of E. Antipodal pair

contains an edge of a convex polygon and its antipodal edge.

- Chord (κ) :- For any two points S and G on the perimeter of a region, the straight line path connecting S and G is called κ(S, G). Thus, any chord of a convex region lies completely inside it.

- Chord-ratio (λ) :- For any two points S and G on the perimeter of a region, the ratio of ρ(S, G) and κ(S, G) is called λ(S, G). Maximum chord-ratio over all points S and G on the boundary of a region R is called the Maximum chord-ratio of R and is denoted by λ_{max}(R). For example, Maximum chord-ratio for a square is 2.

- Chord-angle (θ) :- For any two points S and G on the perimeter of a convex polygon, the angle between κ(S, G) and the edge containing G in the direction of ρ(S, G) is called θ(S, G).

 If G is a vertex of the polygon, it is shared by two edges. In such cases, traversing from S to G in the direction of ρ(S, G), the first edge containing G is to be chosen for finding θ(S, G).

 If G is perimetric image of S, there are two values of θ(S, G) :

 θ_{cw}(S, G) in clockwise direction from S to G.
 θ_{ccw}(S, G) in counter clockwise direction from S to G.

- Critical chord-angle (θ_{Cr}) :- For any two points S and G on the perimeter of a convex polygon the critical chord-angle is given by

$$\theta_{Cr}(\mathrm{S},\ \mathrm{G}) = \cos^{-1}\left(\frac{\kappa(\mathrm{S},\ \mathrm{G})}{\rho(\mathrm{S},\ \mathrm{G})}\right)$$

 Note that,

$$\theta_{Cr}(\mathrm{S},\ \mathrm{G}) = \theta_{Cr}(\mathrm{G},\ \mathrm{S}) \text{ and } 0^o < \theta_{Cr} < 90^o$$

Definition : *A convex region R, having cost index C_R and surrounded by regions having maximum cost index C_S, is a True Obstacle iff, the ratio of C_R and C_S is greater than the Maximum chord-ratio of R ($\lambda_{max}(R)$).*

Consider any convex region R, having cost index C_R and surrounded by regions having maximum cost index C_S such that

$$\left(\frac{C_R}{C_S}\right) > \lambda_{max}(R)$$

Then the cost of straight line path connecting any two points S and G on the perimeter of R is $C_R \times \kappa$, where κ is the chord(S, G).

And cost of the shortest path connecting S and G along the perimeter of R $\leq$ $C_S \times \rho$, where ρ is the peri-dist(S, G). Hence, the ratio of two costs is

$$\begin{aligned} \frac{C_R \times \kappa}{C_S \times \rho} &= \frac{C_R}{C_S} \times \frac{1}{\lambda} \\ &\geq \frac{C_R}{C_S} \times \frac{1}{\lambda_{max}} \end{aligned}$$

Now, since $\frac{C_R}{C_S} > \lambda_{max}$, this ratio is always greater than unity. Hence, the cost of path through region is always greater. For points not on the perimeter, the ratio will be even greater. Hence, for any start and goal points outside R, the minimum cost path connecting start and goal points cannot enter R. Hence, it is a **True Obstacle**. Therefore, for any convex region the condition under which it becomes True Obstacle can be calculated by the geometry of the region itself.

2.2 Maximum Chord-ratio Theorem

Theorem : *For any pair of points S and G on the perimeter of a convex polygon R, chord-ratio ($\lambda(S, G)$) to be maximum ($\lambda_{max}(R)$), it is necessary that*

1. *Neither S nor G can be a vertex.*
2. $\theta(S, G) = \theta(G, S)$.
3. $\theta(S, G) \geq \theta_{Cr}(S, G)$.

 In addition,

4. *if* $\theta(S, G) > \theta_{Cr}(S, G)$, $\rho(S, G) = \rho_{max}(R)$.

5. *if* $\theta(S,\ G) = \theta_{Cr}(S,\ G)$,

 There is a set of parallel chords with the same chord-ratio and one of these chords is between two perimetric image points.

Proof :

Proposition 1 : Consider S as any vertex of a convex polygon R shared by two edges e1 and e2 of R and G as any point on any other edge e3 of R. ϕ_1, ϕ_2 and ϕ_3 are the angles between chord(S, G) and e1, e2, e3 respectively (Fig. 2.2).

$$\text{S being a vertex, } \phi_2 > \phi_1 \tag{2.1}$$

Consider points S′ and G′ on edges e1 and e3 at a distance x from S and G respectively such that $\rho(S', G') = \rho(S, G)$.

$$\begin{aligned} \text{Then, } \Delta' &= \kappa'^2 - \kappa^2 = 2x^2[1 - \cos(\phi_1 - \phi_3)] + 2\kappa x(\cos\phi_3 - \cos\phi_1) \\ \frac{d\Delta'}{dx} &= 2x[1 - \cos(\phi_1 - \phi_3)] + \kappa(\cos\phi_3 - \cos\phi_1) \end{aligned}$$

$$\text{As } x \to 0,\ \frac{d\Delta'}{dx} = \kappa(\cos\phi_3 - \cos\phi_1) \tag{2.2}$$

Now consider points S″ and G″ on edges e2 and e3 at a distance x from S and G respectively such that $\rho(S'', G'') = \rho(S, G)$.

$$\text{Then, as } x \to 0,\ \frac{d\Delta''}{dx} = \kappa(\cos\phi_2 - \cos\phi_3) \tag{2.3}$$

From equations (2.1), (2.2) and (2.3) there are three cases,

1. $\cos\phi_3 \geq \cos\phi_1 > \cos\phi_2$, making $\frac{d\Delta''}{dx}$ negative.
2. $\cos\phi_1 > \cos\phi_3 > \cos\phi_2$ making $\frac{d\Delta'}{dx}$ and $\frac{d\Delta''}{dx}$ negative.
3. $\cos\phi_1 > \cos\phi_2 \geq \cos\phi_3$ making $\frac{d\Delta'}{dx}$ negative.

Thus, $\frac{d\Delta}{dx}$ is always negative in some direction. Hence, chord-ratio cannot be maximum at a vertex.

□

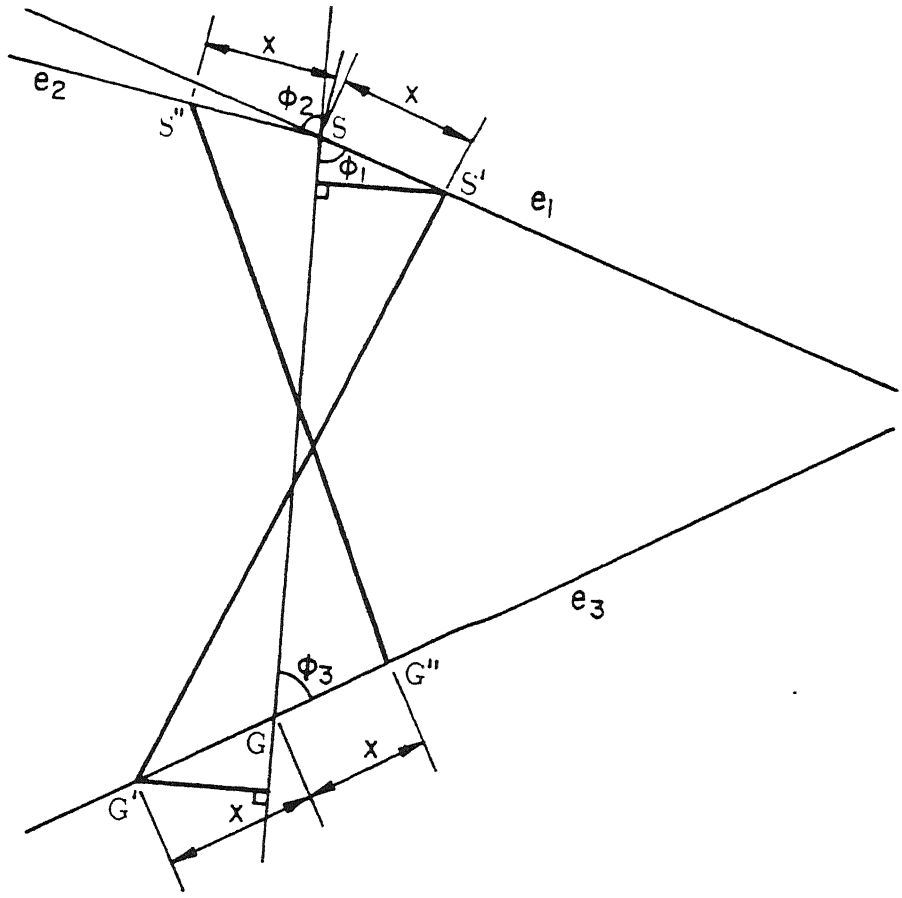
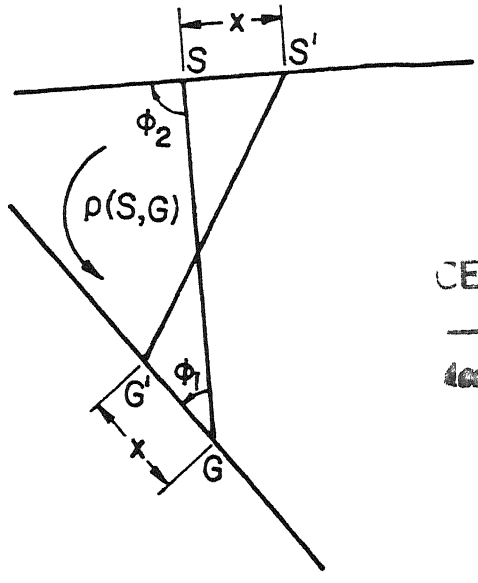


Figure 2.2: Proposition 1 - λ_{max}(R) cannot occur at a vertex of Region R - For $x \rightarrow 0, \lambda$(S, G) is smaller than at least one of λ(S', G') and λ(S'', G'').



CENTRAL LIBRARY
I. I. T. KANPUR
Acc. No. A 117813

Figure 2.3: Proposition 2 - λ_{max} cannot occur at a pair of points S and G, if θ(S, G) $\neq \theta$(G, S) - if θ(G, S) > θ(S, G), λ(S', G') > λ(S, G), for $x \rightarrow 0$. Similar is the case when θ(S, G) > θ(G, S).

Proposition 2 : Consider any two boundary points S and G on a convex polygon R, such that $\theta(G, S) > \theta(S, G)$ i. e., $\phi_2 > \phi_1$ (Fig. 2.3). Without loss of generality, let the peri-dist ($\rho(S, G)$) be measured in ccw direction. Consider two boundary points S′ and G′ at a distance x from S and G respectively along the boundary of R, such that

$$\rho(S', G') = \rho(S, G) \text{ and } \theta(G', S') < \theta(G, S)$$

$$\text{Hence,} \quad \kappa'^2 - \kappa^2 = 2x^2[1 - \cos(\phi_2 - \phi_1)] - 2\kappa x(\cos\phi_1 - \cos\phi_2)$$

$$\text{Since, } \phi_2 > \phi_1 \quad \text{for } x \in \left(0, \frac{\kappa(\cos\phi_1 - \cos\phi_2)}{1 - \cos(\phi_2 - \phi_1)}\right) \quad \text{which is a non-empty set,}$$

$$\kappa'^2 - \kappa^2 < 0$$

$$\text{OR } \lambda(S', G') > \lambda(S, G)$$

Hence, if $\theta(S, G) < \theta(G, S)$, $\lambda(S, G)$ cannot be maximum. Similarly, if $\theta(S, G) > \theta(G, S)$, $\lambda(S, G)$ cannot be maximum. Hence, $\theta(S, G) = \theta(G, S)$ is a necessary condition for $\lambda(S, G)$ to be maximum.

□

Proposition 3 : Consider any two boundary points S and G on a convex polygon R (Fig. 2.4). Without loss of generality, let the peri-dist ($\rho(S, G)$) be measured in ccw direction. Consider a boundary point G′ in the direction of reducing peri-dist, i. e., $\rho(S, G') = \rho(S, G) - x = \rho - x$

$$\text{Now,} \quad \kappa'^2 = \kappa^2 + x^2 - 2x\kappa\cos\theta$$

$$\text{Hence,} \quad \Delta = \lambda'^2 - \lambda^2$$

$$= \left(\frac{\rho^2 + x^2 - 2\rho x}{\kappa^2 + x^2 - 2x\kappa\cos\theta}\right) - \left(\frac{\rho^2}{\kappa^2}\right)$$

As Denominator $= \kappa^2$ $\kappa'^2 =$ positive,

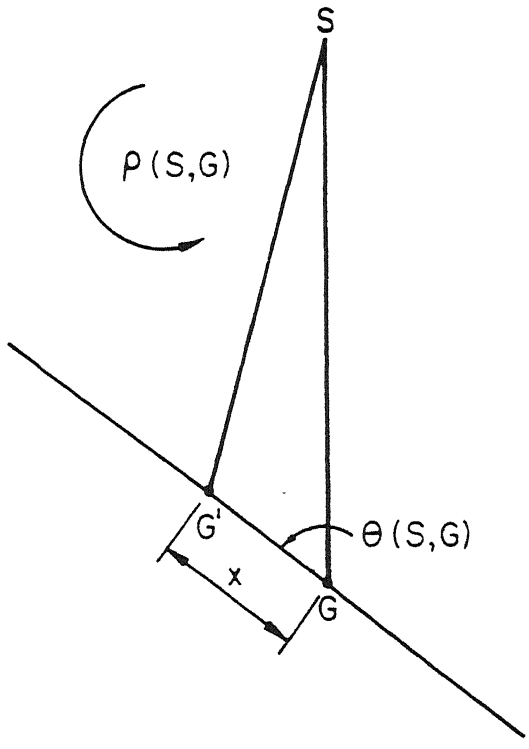


Figure 2.4: Proposition 3 - $\lambda_{max}(\mathrm{R})$ cannot occur at a pair of points S and G, if $\theta(\mathrm{S,\ G}) < \theta_{Cr}(\mathrm{S,\ G})$ - In such a case, we can always choose a point G′ in the direction of reducing peri-dist as shown, such that $\lambda(\mathrm{S,\ G'}) > \lambda(\mathrm{S,\ G})$.

$$\begin{aligned} \Delta_1 &= \text{Numerator} = x^2(\kappa^2 - \rho^2) + 2\rho\kappa x(\rho\cos\theta - \kappa) \\ \frac{d\Delta_1}{dx} &= 2x(\kappa^2 - \rho^2) + 2\rho\kappa(\rho\cos\theta - \kappa) \\ \text{As } x \to 0,\ \frac{d\Delta_1}{dx} &= 2\rho\kappa(\rho\cos\theta - \kappa) \\ &= \text{positive} \quad \text{for } \rho\cos\theta > \kappa \text{ i. e., } \theta < \theta_{Cr}. \end{aligned}$$

Hence, if $\theta(\mathrm{S,\ G}) < \theta_{Cr}(\mathrm{S,\ G})$, $\lambda(\mathrm{S,\ G})$ cannot be maximum and $\theta(\mathrm{S,\ G}) \geq \theta_{Cr}(\mathrm{S,\ G})$ is a necessary condition for $\lambda(\mathrm{S,\ G})$ to be maximum.

□

Proposition 4 : Consider any two boundary points S and G on a convex polygon R (Fig. 2.5). Without loss of generality, let the peri-dist ($\rho(\mathrm{S,\ G}) < \rho_{max}(\mathrm{R})$) be measured in ccw direction. Consider a boundary point G′ in the direction of increasing peri-dist, i. e., $\rho(\mathrm{S,\ G'}) = \rho(\mathrm{S,\ G}) + x = \rho + x$

$$\text{Now,} \quad \kappa'^2 = \kappa^2 + x^2 + 2x\kappa\cos\theta$$

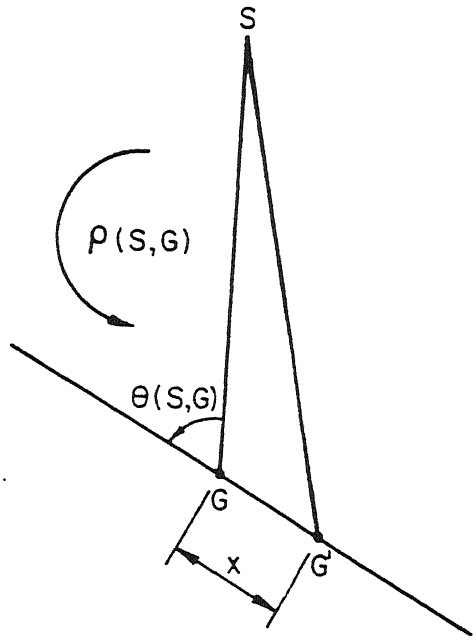


Figure 2.5: Proposition 4 - Given θ(S, G) > θ_{Cr}(S, G), λ_{max}(R) cannot occur at the pair of points S and G if ρ(S, G) < ρ_{max}(R) - In such a case, we can always choose a point G′ in the direction of increasing peri-dist as shown, such that λ(S, G′) > λ(S, G).

$$\text{Hence,} \quad \Delta = \lambda'^2 - \lambda^2$$
$$= \left(\frac{\rho^2 + x^2 + 2\rho x}{\kappa^2 + x^2 + 2x\kappa\cos\theta}\right) - \left(\frac{\rho^2}{\kappa^2}\right)$$

As Denominator $= \kappa^2$ $\kappa'^2 =$ positive,

$$\Delta_1 = \text{Numerator} = x^2(\kappa^2 - \rho^2) + 2\rho\kappa x(\kappa - \rho\cos\theta)$$
$$\frac{d\Delta_1}{dx} = 2x(\kappa^2 - \rho^2) + 2\rho\kappa(\kappa - \rho\cos\theta)$$
$$\text{As} \quad x \to 0, \frac{d\Delta_1}{dx} = 2\rho\kappa(\kappa - \rho\cos\theta)$$
$$= \text{positive} \quad \text{for } \rho\cos\theta < \kappa \text{ i. e., } \quad \theta > \theta_{Cr}.$$

Hence, if θ(S, G) > θ_{Cr}(S, G) and ρ(S, G) < ρ_{max}(R), λ(S, G) cannot be maximum and if θ(S, G) > θ_{Cr}(S, G), ρ(S, G) = ρ_{max}(R) is a necessary condition for λ(S, G) to be maximum.

□

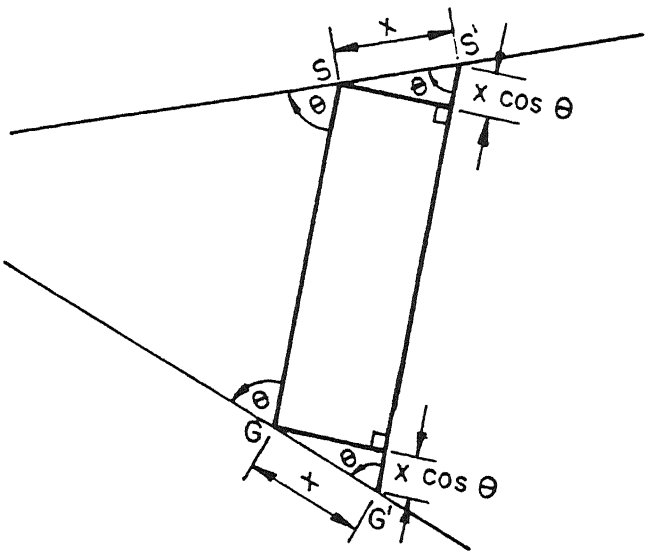


Figure 2.6: Proposition 5 - If θ(S, G) = θ(G, S) = θ_{Cr}(S, G), there is a set of parallel chords with same chord-ratio. If this set does not include the chord between two perimetric image points, λ_{max}(R) cannot occur at this pair of points.

Proposition 5 : Consider any two boundary points S and G on a convex polygon R such that θ(S, G) = θ(G, S) = θ_{Cr}(S, G). Without loss of generality, let the peri-dist (ρ(S, G) < ρ_{max}(R)) be measured in ccw direction (Fig. 2.6). Consider two boundary points S′ and G′ at a distance x from S and G respectively, such that chord(S′, G′) is parallel to chord(S, G).

$$\begin{aligned} \text{Now} \quad \lambda' &= \frac{\rho'}{\kappa'} = \frac{\rho + 2x}{\kappa + 2x\cos\theta} \\ &= \frac{\rho}{\kappa}\left(\frac{\rho + 2x}{\rho + 2x}\right) = \lambda, \quad \text{since,} \quad \rho + 2x \neq 0. \end{aligned}$$

Thus, for a series of parallel chords having $\theta = \theta_{Cr}$, the increase in peri-dist and the increase in chord are proportional so as to maintain the chord-ratio constant. But, if these chords do not include the chord for which peri-dist is maximum, a chord with an end point on a vertex must be included in the series and referring to the proposition (1) above, peri-dist cannot be maximum at a vertex. Hence, this series of chords must include the chord of maximum peri-dist for λ(S, G) to be maximum.

□

2.3 Verifying a True Obstacle

Input : POLYGON, a convex polygon of structure type given below.
Objective : Finding maximum chord-ratio (λ_{max}) for a convex polygon.
Data Structure :

1. VTX : Used for all points : X, Y co-ordinates.
2. EDGE : Used for all line segments.
 - V1, V2 : represent the end points of the line segment.
 - LENGTH : represents the length of the line segment.
 - Edge is associated with a direction from V1 to V2.
 - Line representation of type $\frac{x-x0}{l} = \frac{y-y0}{m} = t$ is used.
3. POLYGON : Used for the convex polygon.
 - NUMEDGE : represents the total number of edges of the polygon.
 - EDGE : array containing all edges of the polygon. [0, ..., numedge − 1].
 - ρ_{max} : contains maximum peri-dist (perimeter/2).
 - λ_{max} : to be calculated by the algorithm.

True Obstacle Algorithm :

1. EDGE = first edge of polygon.
2. IMAGE = antipodal edge of EDGE,
 shorten IMAGE to make IMAGE·V1 = perimetric image of EDGE·V1.
3. MIN-CHORD = ρ_{max}.
4. while (IMAGE ≠ first edge of polygon)

```
    {
      if EDGE and IMAGE are parallel
        {
          de = signed distance of EDGE·V1 from normal dropped on
```

```
                                EDGE from IMAGE·V1.
    if (de > 0 AND EDGE·LENGTH > de/2 AND IMAGE·LENGTH > de/2)
      the pair contains local min, locate, compare with existing MIN-CHORD,
      if smaller, store in MIN-CHORD.
  }
else
  {
    Find intersection point in terms of de and di,
    de = signed distance of intersection point from EDGE·V1,
    di = signed distance of intersection point from IMAGE·V1,
    if ((de + di) > 0 AND EDGE·LENGTH > (de + di)/2 AND
                                IMAGE·LENGTH > (de + di)/2)
      the pair contains local min, locate, compare with existing MIN-CHORD,
      if smaller, store in MIN-CHORD.
  }
Moving to next antipodal pair.
   if (EDGE·LENGTH < IMAGE·LENGTH)
   {
     EDGE = the edge next to current EDGE in the
                edge sequence of the polygon,
     Shorten IMAGE to make IMAGE·V1 = perimetric image of EDGE·V1.
   }
   else if (EDGE·LENGTH > IMAGE·LENGTH)
          {
            IMAGE = the edge next to current IMAGE in the
                       edge sequence of the polygon,
            Shorten EDGE to make EDGE·V1 = perimetric image of IMAGE·V1.
          }
        else
          {
            EDGE = the edge next to current EDGE in the
                       edge sequence of the polygon,
            IMAGE = the edge next to current IMAGE in the
                       edge sequence of the polygon.
```

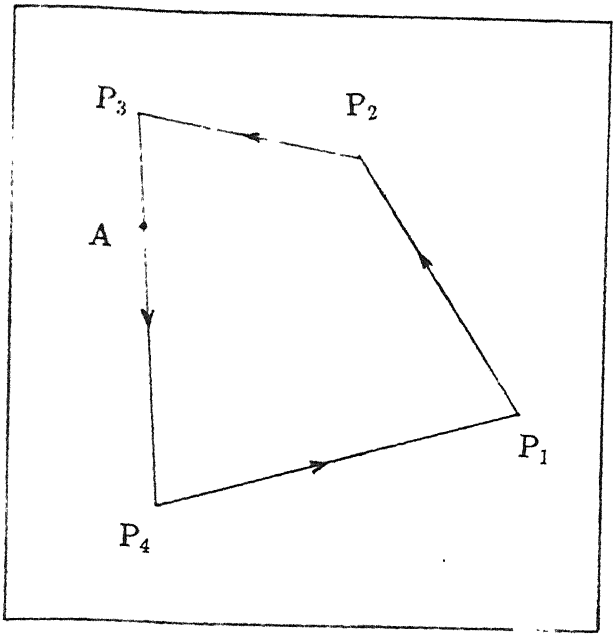


Figure 2.7: Example polygon - Finding λ_{max} of a convex polygon.

}

}

5. $\lambda_{max} = \rho_{max}$/MIN-CHORD.

Example : For the example polygon (Fig. 2.7), initially, EDGE = edge(P_1P_2), IMAGE = (P_3P_4) and after shortening the IMAGE, IMAGE = AP_4. We then locate points S and G (non-vertices) on EDGE and IMAGE respectively, such that θ(S, G) = θ(G, S), if exist, and store distance SG in MIN-CHORD. Then, move to next antipodal pair of edges. Here, IMAGE.LENGTH < EDGE.LENGTH giving IMAGE = edge(P_4P_1) and EDGE gets shortened. This is repeated until all edges are scanned fully. If at any stage, distance SG < existing MIN-CHORD, store distance SG in MIN-CHORD. At the end MIN-CHORD contains the minimum of chords for all perimetric image points of the polygon. Maximum chord-ratio is then calculated.

Complexity Analysis : The algorithm starts with the antipodal pair for first edge of the polygon. These antipodal edges are either parallel or non-parallel, requiring constant time for testing. Next antipodal pair is then selected for testing.

As the algorithm moves to the next antipodal pair, at least one side (either IMAGE or EDGE) is advanced to the next edge in the edge sequence of the polygon. Further, algorithm selects antipodal pairs moving in only one direction (ccw or cw from the first edge of polygon), till it reaches the antipodal edge of the first edge, each edge is scanned only once. Hence, for a polygon of N vertices the algorithm is O(N) complex.

Proof of Correctness : The algorithm scans each edge of the polygon completely moving in one direction (ccw or cw starting from first edge). There are only two cases possible for antipodal pairs : the edges are either parallel or non-parallel. For both cases the algorithm finds a local maximum chord-ratio, if it exists. The maximum of all such local maximum chord-ratios is selected as maximum chord-ratio (λ_{max}) for the polygon. Hence, by the maximum chord-ratio theorem, the result obtained by the algorithm is λ_{max}.

2.4 Genetic Algorithms

Genetic Algorithms, or GAs, as they have been called, are effective search and optimization techniques. GAs are very different than traditional optimization techniques. GAs simulate the rules of natural genetics, in order to find out global optimum, analogous to "fittest in species". To solve an optimization problem by GA, we have to create a random population of points and assign their "fitness" values according to objective function value. Thereafter the following four operators are applied repeatedly to arrive at optimal solution :

1. Reproduction
2. Cross-over
3. Mutation
4. Evaluation of population

Reproduction creates the next generation in such a way that individuals with higher fitness get more copies compared to those with lesser fitness. Cross-over creates

new children from the parent generation, exploiting similarity among parents. Mutation is the operator which ensures diversity in the population and guards against premature convergence. Finally, evaluation of population is necessary to guide the reproduction in next generations.

GAs have been effectively used for a wide variety of problems in past and are able to get solutions which are not possible by traditional optimization techniques. In an effort to verify the results of our algorithm, which finds the maximum chord-ratio for a convex polygon, we compared it with a GA implementation [Mukerjee, et al.] and obtained similar results.

2.5 Results

After implementing the algorithm we tested it for randomly generated convex polygons. The time for finding the maximum chord-ratio of the polygon increases linearly with the number of vertices (Fig. 2.8), which is in accordance with the complexity analysis. We compared the results obtained by our algorithm with the chord-ratios computed by applying GA for a large number of convex polygons, and found similar values. Some sample convex poygons with perimetric image points and chords of maximum chord-ratio are shown in Fig. 2.9 and 2.10.

The number of generations and population size used in GA for testing purposes were 50, and chord-ratios found were similar upto sixth decimal place.

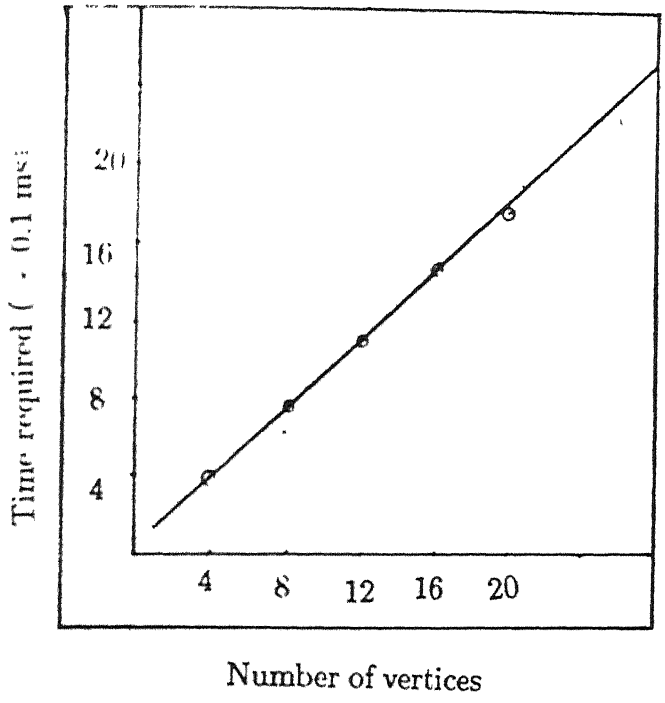


Figure 2.8: Graph of time complexity for the true obstacle algorithm. The linear nature complies with complexity analysis result.

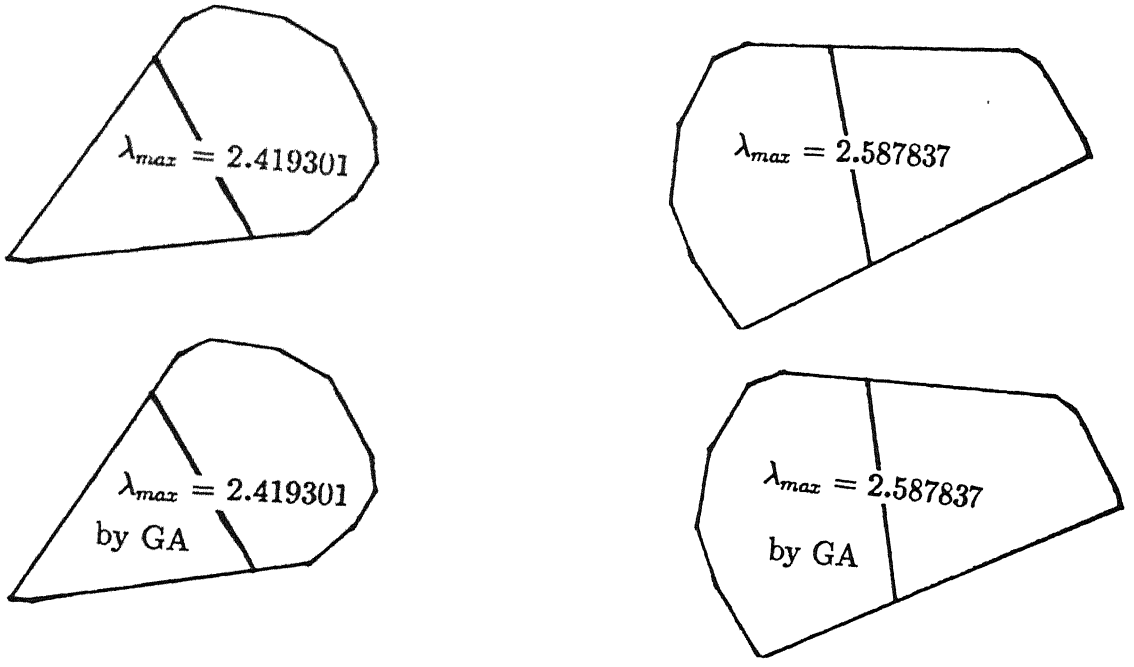


Figure 2.9: Some sample randomly generated convex polygons tested by the true obstacle algorithm. Chord shown is between perimetric image points for which the chord-ratio is maximum (λ_{max}(R)).

CENTRAL LIBRARY
I I T KANPUR

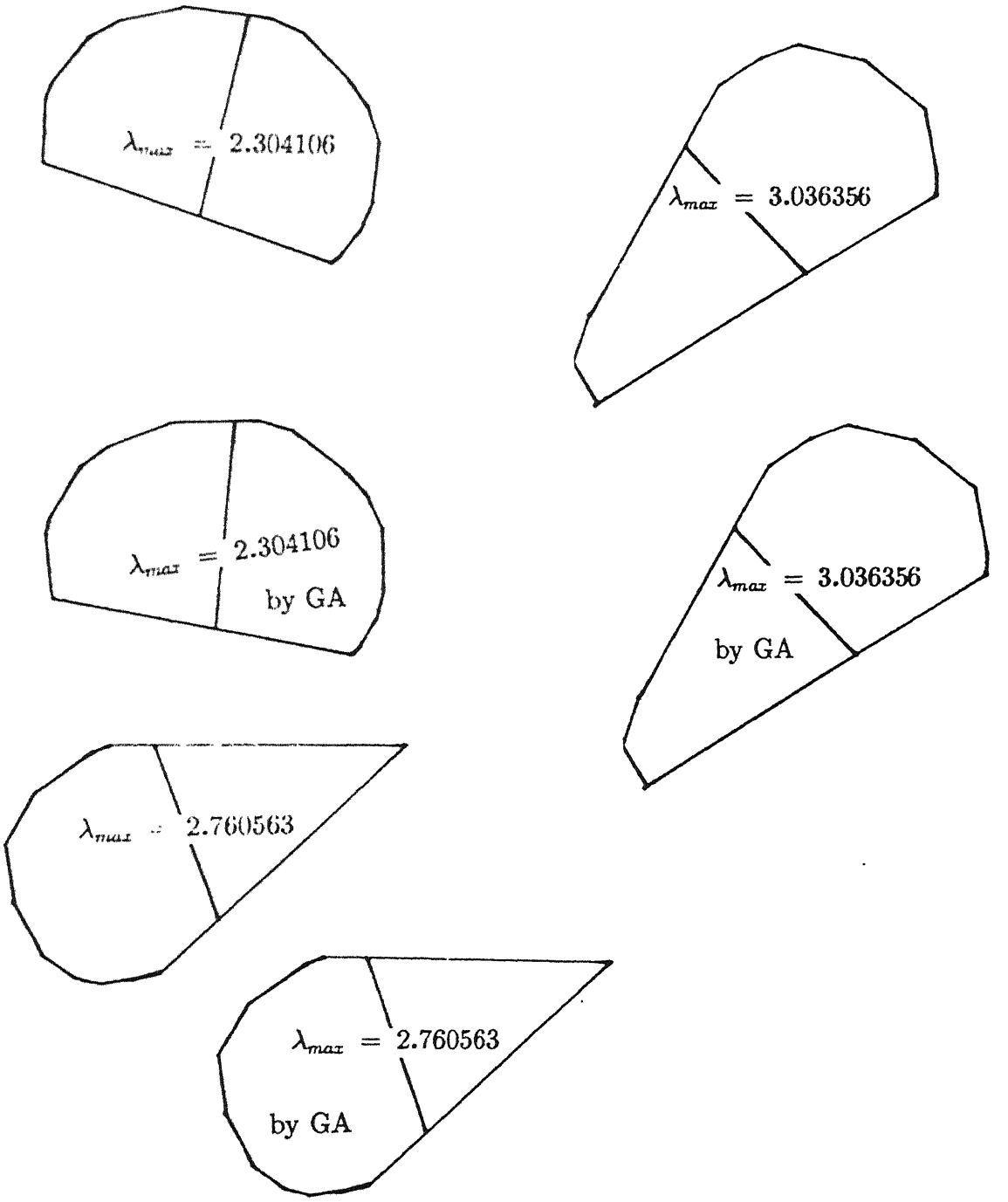


Figure 2.10: Some more sample convex polygons tested by the true obstacle algorithm. Chord shown is between perimetric image points for which the chord-ratio is maximum ($\lambda_{max}(R)$).

Chapter 3

Snell's Law In Path Planning

We have seen how a map can be divided into polygonal regions having uniform isotropic costs. Moving in such uniform cost regions is analogous to moving of light rays in media of uniform light speeds. When an optimum path crosses a region boundary it obeys Snell's Law. In this chapter we discuss how to utilize such optical analogies in path planning.

3.1 Minimum Cost Path

If the minimum cost path between any two points lies completely inside a uniform cost convex region, it is a straight line connecting the two points. This is because, in order to minimize the cost of such a path, we have to minimize the path length. Hence, it can be concluded that the minimum cost path between any two points in a map represented as uniform cost regions, is piecewise linear, changing its direction only at region boundaries. We can, thus, formulate the problem of minimizing the cost of a path between two points separated by N regions as follows :

Given N regions R_1, R_2, ..., R_N with uniform costs C_1, C_2, ..., C_N respectively (Fig.3.1), the path {S, P_1, P_2, ..., P_{N-1}, G} from a point S in region R_1 to a point G in region R_N has total length

$$\left[(SP_1) + \sum_{i=1}^{N-2} (P_i\ P_{i+1}) + (P_{N-1}\ G) \right]$$

The total cost of this path is

$$TC = (SP_1\ C_1) + \sum_{i=1}^{N-2} (P_i\ P_{i+1})\ C_{i+1} + (P_{N-1}\ G)\ C_N$$

Here, P_i is a point on the boundary of regions R_i and R_{i+1}, for $i = 1, 2, \ldots, N-1$.

The cost minimization problem is that of selecting the points P_1, P_2, ..., P_{N-1} so as to minimize the total cost (TC) of the path between S and G.

3.2 Snell's Law and Path Planning

Snell's Law states that the path of a light ray passing through a boundary between media A and B with indices of refraction μ_A and μ_B respectively, obeys the relationship

$$\mu_A \sin\theta_A = \mu_B \sin\theta_B$$

where,

θ_A, the angle of incidence, is the ccw angle between the incoming ray and the normal to the region boundary at the point of incidence (Fig. 3.2).

θ_B, the angle of refraction, is the ccw angle between the outgoing ray and the normal to the region boundary at the point of incidence.

Refractive index (μ) of a medium is given by the relation

$$\mu = \frac{\text{speed of light in vacuum}}{\text{speed of light in the medium}}$$

Now, let us find out the behaviour of minimum cost path at a region boundary. Consider two points S in region A with cost C_A and G in region B with cost C_B (Fig. 3.3). Cost of the path from S to G is then given by

$$\begin{aligned} \text{path-cost} &= C_A l_A + C_B l_B \\ &= C_A(d_S^2 + x^2)^{1/2} + C_B[d_G^2 + (a-x)^2]^{1/2} \end{aligned}$$

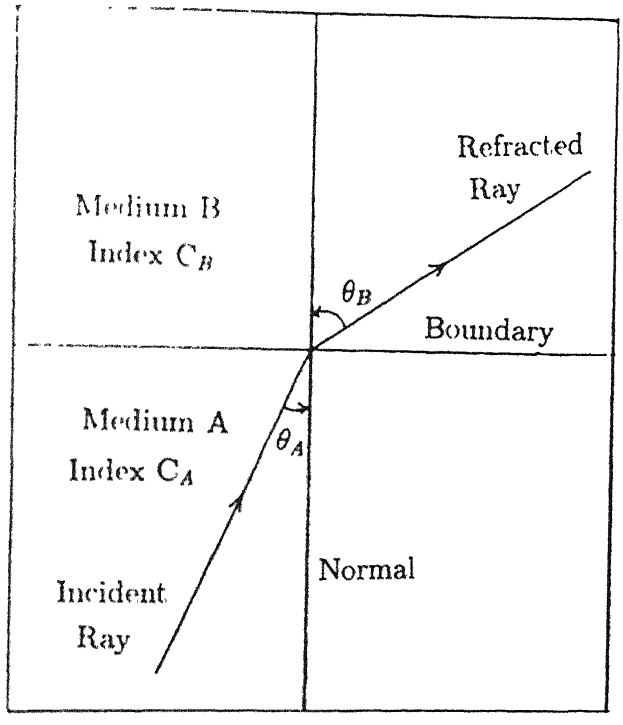


Figure 3.2: Snell's Law of refraction - Light ray refracts at the boundary of media A and B with indices of refraction μ_A and μ_B obeying the ralation $\mu_A \sin\theta_A = \mu_B \sin\theta_B$.

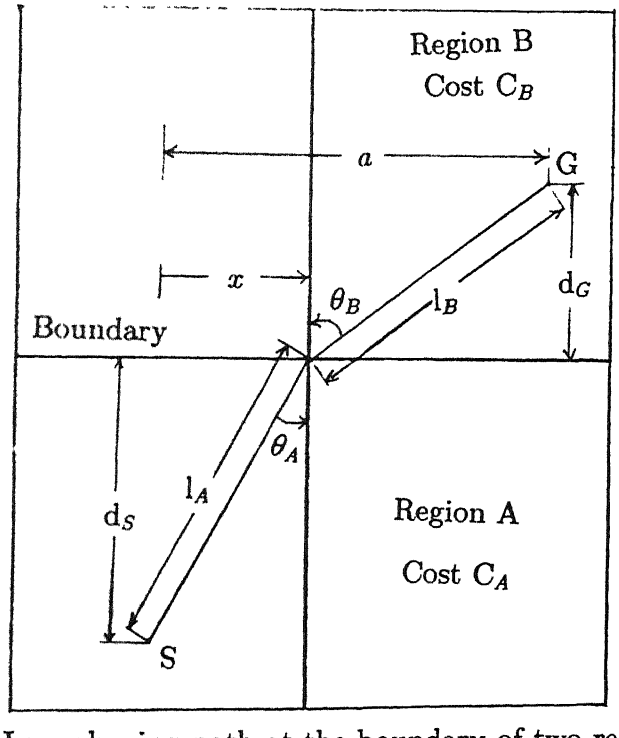


Figure 3.3: Snell's Law obeying path at the boundary of two regions is the minimum cost path.

For minimizing,

$$\begin{aligned} \frac{d}{dx}(\text{path-cost}) &= 0 \\ \text{or} \quad C_A \sin\theta_A &= C_B \sin\theta_B \end{aligned}$$

If we consider costs to be inversely proportional to velocity, this is Snell's Law. Hence, Snell's Law obeying path at the boundary of uniform cost regions is the minimum cost path.

Critical Angle : When light rays enter from a denser medium into a rarer medium, the angle of incidence for which the angle of refraction is 90^o, is called the *critical angle* and refraction in this case is called as *critical refraction*. If the angle of incidence is greater than the critical angle, light rays get totally reflected back into the denser medium. In path planning it simply means that if a minimum cost path crosses the region boundary and there is no angle of incidence less than the critical angle, it does not obey Snell's Law. The critical angle is given by

$$\theta_{cr} = \sin^{-1}(C_B/C_A),\ C_B < C_A.$$

The minimum cost path doesn't always behave similar to the light rays in analogous optical media. Example 1, below, shows a case where light rays cannot reach goal point and the minimum cost path to the goal doesn't obey Snell's Law at region boundary. In examples 2 and 3, paths followed by light rays in analogous optical media are not minimum cost paths.

Example 1 : If the minimum cost path has to enter from a high cost region to a low cost region, sometimes there is no angle of incidence smaller than the critical angle at any of the region boundaries (Fig. 3.4). Hence, no path obeying Snell's Law, can enter the low cost region to reach the goal. For light rays, these boundaries are like mirrors. The minimum cost path (S-P-G) in such cases doesn't obey Snell's Law at region boundaries.

Example 2 : When start point S and goal point G, both lie inside a high cost convex region surrounded by a low cost region (Fig. 3.5), light rays follow a straight line path connecting S and G, lying completely inside the high cost region. In such

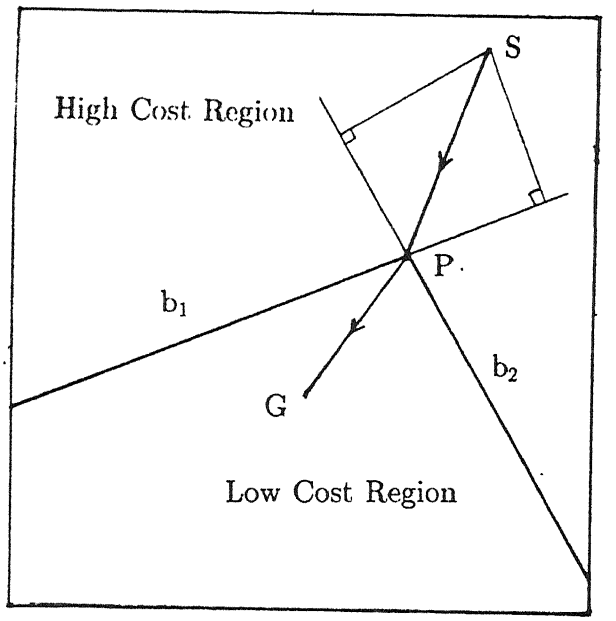


Figure 3.4: Mirror boundaries - Although the minimum cost path is {S, P, G}, light will not be able to enter the low cost region, as the boundaries act like mirros.

a case, it might be more economical to follow a path along the boundary, lying just inside the low cost region. Snell's Law does not suggest such minimum cost paths.

Example 3 : If start point S and goal point G lying in a low cost region are separated by a high cost region such that the straight line path connecting S and G is normal at the high cost region boundaries (Fig. 3.6), Snell's Law obeying path may not be the minimum cost path.

3.3 Multiregion Snell's Law

To compute Snell's Law obeying path from S to G separated by a region boundary (Fig 3.3), we get the quartic equation

$$x^4 + A_3x^3 + A_2x^2 + A_1x + A_0 = 0$$

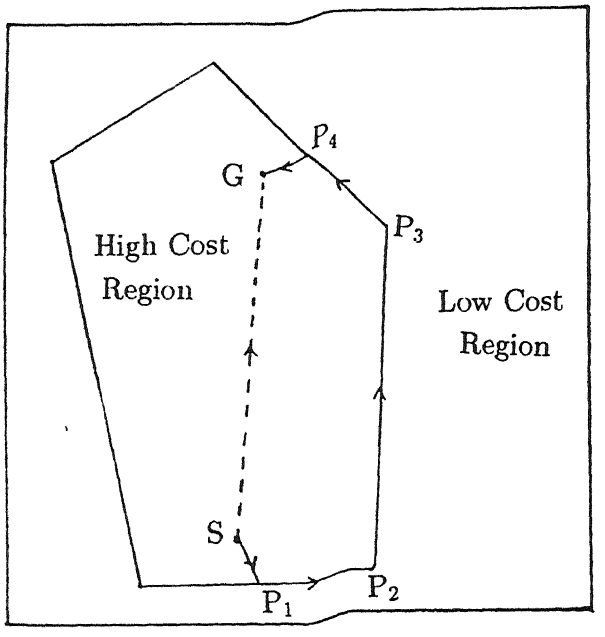


Figure 3.5: The path followed by light (dashed line) has a higher cost than the path $\{S, P_1, P_2, P_3, P_4, G\}$ which follows the contour in the low cost region.

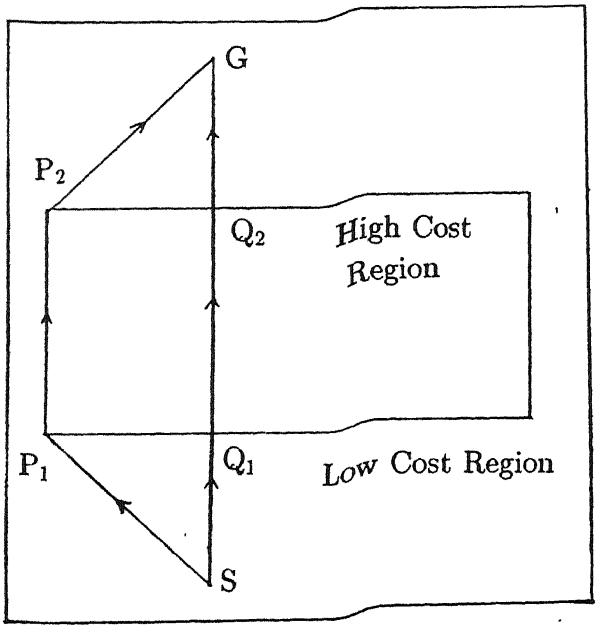


Figure 3.6: The path $\{S, Q_1, Q_2, G\}$ has a higher cost than the path $\{S, P_1, P_2, G\}$.

Where,

$$
\begin{aligned}
A_3 &= -2a \\
A_2 &= \frac{C_A^2(a^2 - d_G^2) - C_B^2(a^2 - d_S^2)}{C_A^2 - C_B^2} \\
A_1 &= \frac{2ad_S^2C_B^2}{C_A^2 - C_B^2} \\
A_0 &= -\frac{a^2d_S^2C_B^2}{C_A^2 - C_B^2}
\end{aligned}
$$

Hence, the complication in Multiregion Snell's Law is that when we write down the equations that represent the local optimality criterion (Snell's Law) at each edge, and then try to solve for the points where the path crosses edges, we get $(N-1)$ quartic equations in $(N-1)$ unknowns, where $(N-1)$ is the length of edge sequence (Fig 3.1) through which the refraction path is known to pass. Elimination among these equations yields a very high degree polynomial (of degree exponential in $(N-1)$ [Mitchell]).

In this chapter, we present an iterative search algorithm to find the Snell's Law obeying path through a given sequence of edges. Basic idea is to cast a ray from S and find out the error at goal (G) end. The next ray is then cast to reduce this error. This is repeated until the error at goal end is less than the specified bound.

Problem : To compute the Snell's Law obeying path from a point S in region R_1 to a point G in region R_N, which crosses the boundary segments $b_1, b_2, \ldots, b_{N-1}$ of regions $R_1, R_2, \ldots, R_N$ with uniform costs $C_1, C_2, \ldots, C_N$ respectively (Fig.3.1).

Nomenclature

Limiting intercepts : These are the points on b1. If a ray is cast from S to limiting-left-intercept, after final refraction, G lies to the right of this ray. Similarly for limiting-right-intercept, G lies to the left of ray. Hence, Snell's Law obeying path from S to G lies between limiting intercepts. During the search, difference between

limiting intercepts is successively reduced to find the required path. Initially these intercepts are the end points of b1.

Errors : If a ray cast from S does not pass through G after refraction through last boundary segment, its deviation from G is the calculated-error. Aim is to reduce this error for next ray cast. Limiting errors are deviations of rays for limiting intercepts. If for any boundary segment, the ray does not intersect it or critical refraction occurs, error value is near ∞, in the algorithm it is represented by HIGH-VALUE. If error is less than the threshold given, search is complete.

3.4 Multiregion Snell's Law Algorithm

Algorithm : Computing the Snell's Law path through a given sequence of boundary segments.
Input : Start point S and Goal point G, given sequence of boundary segments, costs of regions.
Procedure :

1. Initialize the limits.
 limiting-left-intercept = 0.0.
 limiting-right-intercept = length of b1.
 /* Compute limiting-left-error. */
 Cast a ray from S along limiting-left-intercept.
 Compute refraction path resulting from this ray.
 if for any boundary segment, ray does not intersect
 the segment or critical refraction occurs
 limiting-left-error = Negative HIGH-VALUE.
 else if |calculated-error| < Threshold, terminate search.
 else limiting-left-error = calculated-error.
 /* Compute limiting-right-error. */
 Cast a ray from S along limiting-right-intercept.
 Compute refraction path resulting from this ray.

```
        if for any boundary segment, ray does not intersect
          the segment or critical refraction occurs
          limiting-right-error = Positive HIGH-VALUE.
        else if calculated-error < Threshold, terminate search.
            else limiting-right-error = calculated-error.
2. Cast initial ray from S along initial intercept.
    if limiting-left-error == Negative HIGH-VALUE or
      limiting-right-error == Positive HIGH-VALUE
      initial intercept = mean of limiting intercepts.
    else initial intercept = weighted mean of limiting intercepts.
3. do
    {
     Compute refraction path resulting from this ray.
       if for any boundary segment, ray does not intersect
         the segment or critical refraction occurs
         if G is to the right of this ray
           limiting-left-intercept = present intercept.
         else limiting-right-intercept = present intercept.
         Cast next ray from S along
              next intercept = mean of limiting intercepts.
       else if |calculated-error| < Threshold, terminate search.
           else if G is to the right of this ray
                  limiting-left-intercept = present intercept
                  limiting-left-error = calculated error
                else limiting-right-intercept = present intercept
                     limiting-right-error = calculated error
                if limiting-left-error == Negative HIGH-VALUE or
                  limiting-right-error == Positive HIGH-VALUE
                  Cast next ray from S along
                       next intercept = mean of limiting intercepts.
                else Cast next ray from S along
                          next intercept = weighted mean of limiting intercepts.
    }
```

For special cases, like parallel boundary segments, more efficient algorithms can be used. To verify multiregion Snell's Law algorithm, follwing property of parallel boundary segments is used.

Parallel boundary segments : If the boundary segments b_1, b_2, ..., b_{N-1} of regions R_1, R_2, ..., R_N with costs C_1, C_2, ..., C_N are parallel, angle of refraction for one boundary segment becomes angle of incidence for the next boundary segment. Applying Snell's Law at each boundary, we get the relation,

$$C_1 \sin\theta_1 = C_2 \sin\theta_2 = \cdots = C_N \sin\theta_N$$

where θ_i and θ_{i+1} are angles of incidence and refraction respectively, for boundary segments b_i, $i = 1, 2, \ldots, N-1$.

Results : The algorithm is implemented and tested for S and G separated by upto 25 boundary segments. Results for parallel boundary segments were also tested and found matching with the theoretical values. The algorithm converges quickly. Sample cases are given in Fig. 3.7 and 3.8.

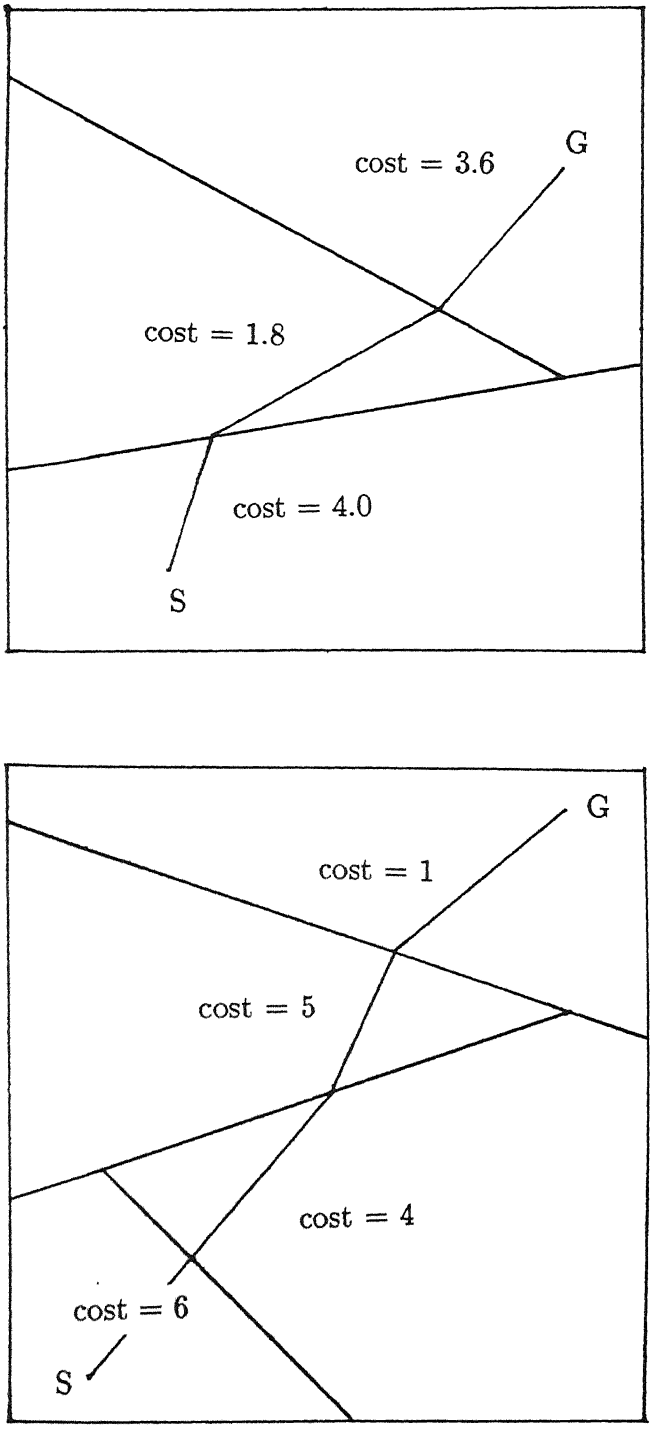


Figure 3.7: To find the minimum cost path between S and G in convex polygonal regions, which is known to pass through a sequence of edges.

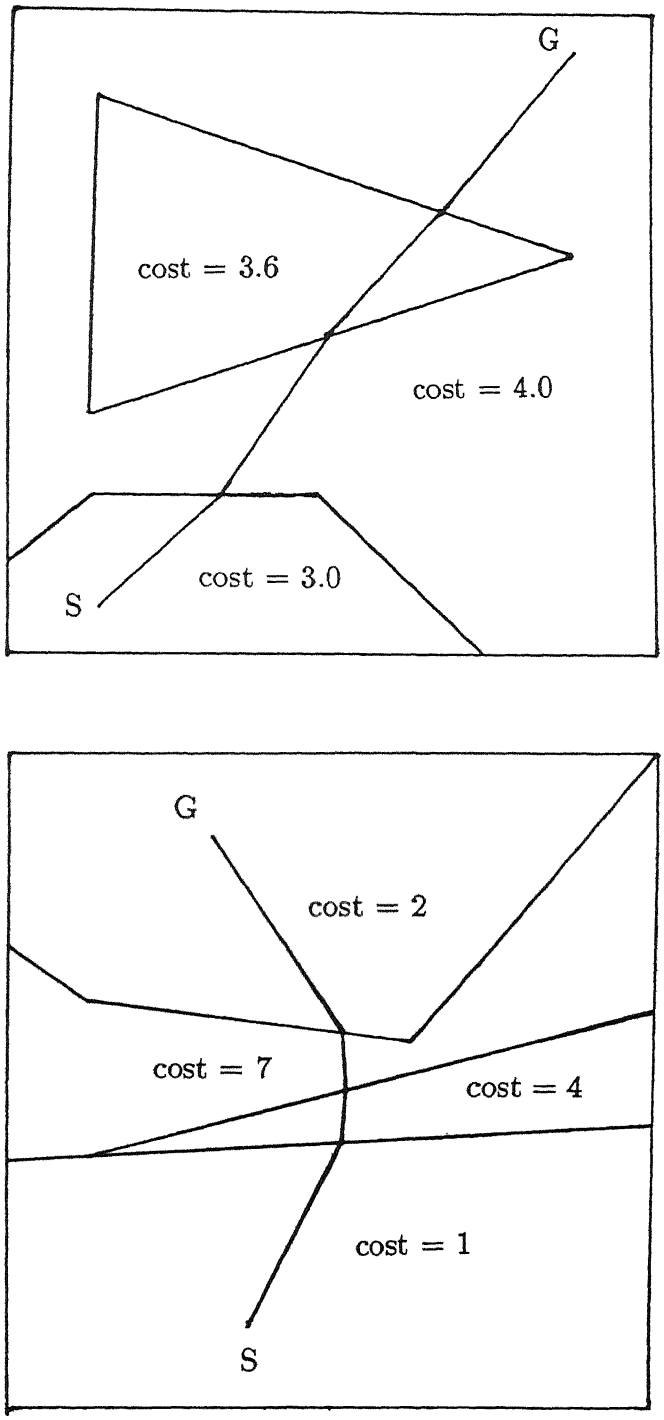


Figure 3.8: To find the minimum cost path between S and G in concave polygonal regions, which is known to pass through a sequence of edges.

Chapter 4

Conclusions

We have seen in brief how a terrain map can be divided into different regions depending upon the negotiability in the terrain. In practice, the negotiability depends on many other factors e.g., time of moving during the day, season of year, presence of other vehicles, etc. All these factors influence the negotiability in the terrain in some or the other way. Also, we have assumed uniform isotropic costs which simplifies the path planning problem greatly. Cosidering all these factors, may ensure real world path planning, and effort in this direction is needed.

The algorithm presented for finding the condition under which a region will become a true obstacle applies to convex polygonal regions. This gives better knowledge of the regions of a map, consequently, simpler algorithms can be applied for path planning. In practice however, we come across terrain maps which cannot be decomposed into convex regions fully. Hence, for completeness, one needs to solve the algorithm to handle concave regions as well. For this, some of the concepts might require modification, e.g., chord for a concave region may not lie completely inside the region.

The optics analogy needs careful study before using it for path planning purposes. The algorithm presented finds the minimum cost path that passes through a given sequence of edges. But the question is : how to select the sequence of edges? Research is needed for selecting the edges to be crossed during navigation through general terrain.

The concept of true obstacles tells which regions should be avoided during path planning. If we know which edges are to be crossed during navigation, the minimum cost path through given sequence of edges can be computed using the algorithm presented. Thus, the two algorithms form tools for the general terrain path planning.

References

Latombe, J. C. (1991). Robot Motion Planning, *Kluwer Academic Publishers*, Boston.

Mitchell, J. S. B. (1988). An algorithmic approach to some problems in terrain navigation, *Arificial Intelligence*, Vol. 37, pp 171-201.

Moravec, H. P. (1989). Sensor Fusion in Certainity Grids for Mobile Robots, *NATO Series*, Vol. F52, pp 253-276.

Mukerjee, A., Sharma, S.P., and Agarwal, R. B. (1994). When an Obstacle is an Obstacle?, Under preparation.

Preparata, F. P., and Shamos, M. I. (1985). Computational Geometry, *Springer*, New York.

Rowe, N. C., and Richbourg, R. F. (1990). An Efficient Snell's Law Method for Optimal-path Planning across Multiple Two-dimensional, Irregular, Homogeneous-cost Regions, *International Journal of Robotics Research*, Vol. 9, No. 6, pp 48-66.